

# Development of Real-Time Adaptive Noise Canceller and Echo Canceller

Jean Jiang, *Member, IEEE*

**Abstract**—In this paper, the adaptive cancellation structure is first developed based on the LMS algorithm and FIR adaptive filtering. Then the novel practical noise and echo cancellation systems are built using the proposed adaptive technique and implemented using TX320TMS67C13 DSKs, which are Texas Instruments' Digital Signal Processing (TI DSP) boards. Although adaptive filtering is an exciting topic in which many real-life applications can be explored [1]-[6], [9], building such a real-time system is often challenging due to the use of theoretical math, advanced DSP knowledge and practical industrial hands-on experience [1],[4]-[6],[9]. Therefore, this paper indicates that it is possible to apply traditional mathematics in adaptive filtering theory to real-time practical DSP systems. With the MATLAB software tool, we can simulate and verify various adaptive filtering designs first. Then, development and implementation of different noise or echo cancellation systems with adaptive filtering techniques can be successfully performed using the floating-point digital signal processor, TX320TMS67C13 DSK. Furthermore, it can be shown that TX320TMS67C13 DSKs with their stereo channels offer more effective and flexible tools for various noise cancellation applications.

**Index Terms**—Adaptive finite impulse response filter, DSP board, echo cancellation, LMS algorithm, noise cancellation.

## I. INTRODUCTION

The adaptive filter techniques play an important role in modern digital processing (DSP) products in areas such as noise cancellation, telephone echo cancellation, equalization of communications channels, biomedical signal enhancement, active noise control, and adaptive control systems.[1]-[3],[9]. Adaptive filters work generally for the adaptation of signal-changing environments, for spectral overlap between noise and the signal, and for unknown or time-varying noise. Adaptive filtering has existed for more than two decades in the research community and is still active. Although great progress has been made, there are still some limits for practical application. One challenging problem is applying theoretical math and advanced DSP knowledge to a practical system using a floating-point TI DSP board so that a more accurate iterated and adaptive solution can be obtained in real-time.

In this paper, Section II gives the derivations of the proposed least mean square (LMS) algorithm and finite impulse response (FIR) adaptive filtering, and verifies them using a real speech waveform. In Section III, the real-time

system set-up is presented first using a TI DSP board and applications of noise cancellation and echo cancellation are demonstrated using the adaptive technique and TI DSP boards. Finally, conclusions are given in Section IV.

## II. WIENER FILTERS AND ADAPTIVE FILTERS

An adaptive filter is a digital filter that has self-adjusting characteristics. It is capable of adjusting its filter coefficients automatically to adapt to the input signal via an adaptive algorithm. Therefore, adaptive filters have many real applications in system modeling, noise cancellation, adaptive differential PCM (ADPCM), echo cancellation and so on. Let us start from a Wiener filter solution with the steepest descent algorithm, then develop the least mean square (LMS) algorithm, and finally extend a single coefficient LMS adaptive FIR filter to the standard LMS adaptive FIR filter. In addition, MATLAB simulations can be performed to verify the theoretical development.

Fig. 1 shows a Wiener filter for noise cancellation, where a single coefficient filter is used for illustration; that is,  $y(n) = wx(n)$ .  $w$  is the adaptive coefficient and  $y(n)$  is the Wiener filter output, which approximates noise  $n(n)$  in the corrupted signal.

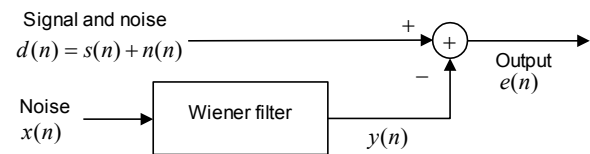


Fig. 1 Wiener filter for noise cancellation.

The enhanced (restored) signal  $e(n)$  is given by

$$e(n) = d(n) - wx(n) \quad (1)$$

Taking statistical expectation of the square of error leads to a quadratic function

$$J = \sigma^2 - 2wP + w^2R \quad (2)$$

where  $R^{-1}$  is the mean square error (MSE) function or output power. When it is minimized, the noise power is maximally reduced. Since  $\sigma^2 = E[d^2(n)]$  (auto-correlation or power of

Manuscript received.

Jean Jiang is with the College of Engineering and Technology, Purdue University North Central, 1401 South U.S. Highway 421, Westville, IN 46391, USA (e-mail: jjiang@pnc.edu).

the corrupted signal),  $P = E[d(n)x(n)]$  (cross-correlation), and  $R = E[x^2(n)]$  are constants,  $J$  is a quadratic function of  $w$  which may be shown in Fig. 2:

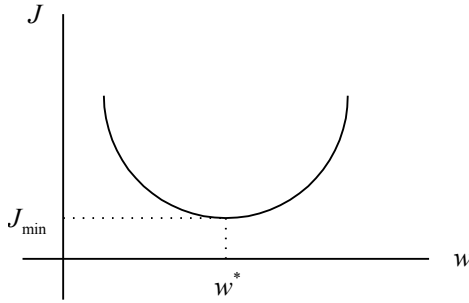


Fig. 2 MSE quadratic function.

The best coefficient (optimal)  $w^*$  is unique and at the location corresponding to the minimum MSE error  $J_{\min}$ . By taking derivative of  $J$  and setting it to zero leads the solution as

$$w^* = R^{-1}P \quad (3)$$

Our goal is to develop an adaptive filter that seeks to minimize the MSE function (i.e. objective function) by iteratively adjusting its parameters (such as its impulse response coefficients) to achieve the design specifications. However, solving the Wiener solution requires a lot of computations, including matrix inversion ( $R^{-1}$ ) for a multiple-tap FIR structure.

Next, the steepest descent algorithm that is capable of minimizing the MSE sample by sample to locate the filter coefficient(s) is introduced as follows:

$$w_{n+1} = w_n - \mu \frac{dJ}{dw} \quad (4)$$

$\mu$  = constant controlling speed of convergence or convergence factor and  $dJ/dw$  is the gradient of the MSE function. The steepest descent method is effective since it can be used to avoid taking the matrix inverse of  $R$  (for  $N$  filter coefficients), which may become to be ill-conditioned. Again, it can be proven [1] that the optimal coefficient can be approached using iterations. To develop the LMS (Least Mean Square) algorithm [1] in terms of the data samples in processing, the gradient  $dJ/dw$  can be approximated by its instantaneous value, that is,

$$\frac{dJ}{dw} = 2(d(n) - wx(n)) \frac{d(d(n) - wx(n))}{dw} = -2e(n)x(n) \quad (5)$$

Substituting the instantaneous gradient  $dJ/dw$  to the steepest descent algorithm, the LMS algorithm for updating a single coefficient is achieved as

$$w_{n+1} = w_n + 2\mu e(n)x(n) \quad (6)$$

Finally, by omitting iteration time index  $n$  and extending one coefficient filter to an- $N$  coefficient filter, the standard LMS algorithm is obtained and listed in Table I.

Table I. LMS adaptive FIR filter with  $N$  filter coefficients.

|   |
|---|
| (1) Initialize $w(0), w(1), \dots, w(N-1)$ to arbitrary values  |
| (2) Read $d(n), x(n)$ , and perform digital filtering<br>$y(n) = w(0)x(n) + w(1)x(n-1) + \dots + w(N-1)x(n-N+1)$      |
| (3) Compute the output error<br>$e(n) = d(n) - y(n)$  |
| (4) Update each filter coefficient using the LMS algorithm for $i = 0, \dots, N-1$<br>$w(i) = w(i) + 2\mu e(n)x(n-i)$ |

To illustrate functionality of the developed adaptive filter, a MATLAB implementation for noise cancellation is provided. The principle of noise cancellation is shown in Fig. 3, in which ADC and DAC stand for Analog-to-Digital Converter and Digital-to-Analog Converter, respectively.

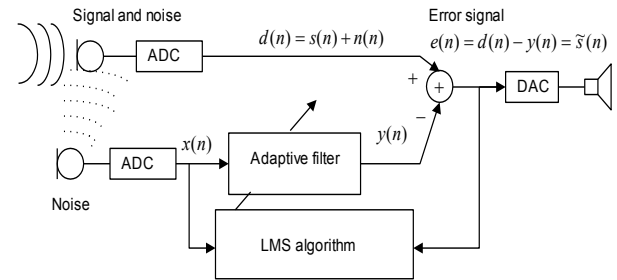


Fig. 3 Noise canceller using an adaptive filter.

The noise cancellation system in Fig. 3 is assumed to have the following specifications: the sampling rate is 8,000 Hz and the speech is corrupted by Gaussian noise with a power of 1 and delayed by 5 samples from the noise reference. The noise reference contains Gaussian noise with a power of 1; an adaptive finite impulse response (FIR) filter is used to remove the noise; the number of FIR filter coefficients is 21, and the convergence factor for the LMS algorithm is chosen to be 0.01. The speech waveforms and speech spectral plots for original speech, corrupted speech, reference noise, and clean speech are plotted in Figure 4, respectively. It is observed that the enhanced speech waveform and spectrum are very close to the original ones. The LMS algorithm converges after approximately 400 iterations. Particularly, in time domain, the waveforms of the original speech  $s(n)$  and the clean speech (error signal  $e(n)$ ) are almost identical after 400 samples when the LMS algorithm converges. The corrupted signal  $d(n)$  and the noise  $x(n)$  are also plotted to show the speech signal has been severely polluted. However, the speech has been well

restored and the mixed noise has been successfully removed after adaptive filtering. The same conclusion can be reached in the frequency domain, where the original speech spectrum and the recovered (clean) spectrum are almost identical. In summary, the adaptive method is a very effective approach for noise canceling.

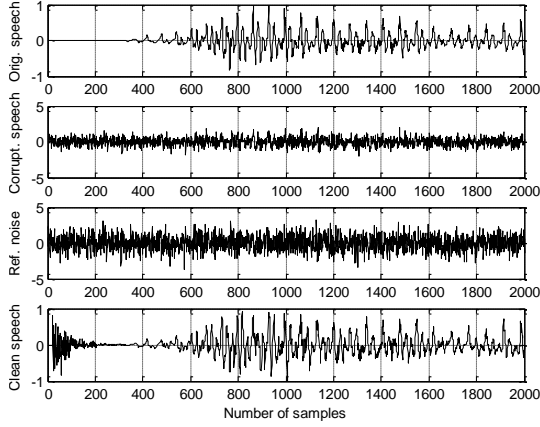


Fig. 4 (a) Original speech, corrupted speech, reference noise and clean speech.

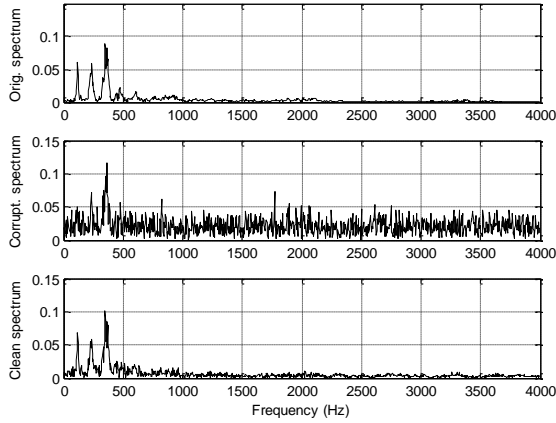


Fig. 4(b) Spectra of Original, corrupted, and clean speech.

### III. REAL-TIME ADAPTIVE SYSTEMS

#### A. System Set-up

For real-time adaptive realizations, a TI TMS320C6713 floating-point processor [8]-[9] is selected instead of a fixed-point digital signal processor (DSP) to avoid the implementation challenges, so that the adaptive filtering solutions can be easily and accurately obtained using C. A single DSP board setup and program segment for verifying input and output signals are shown in Fig. 5a and 5b, respectively, where the sampling rate used is 8,000 samples per second.

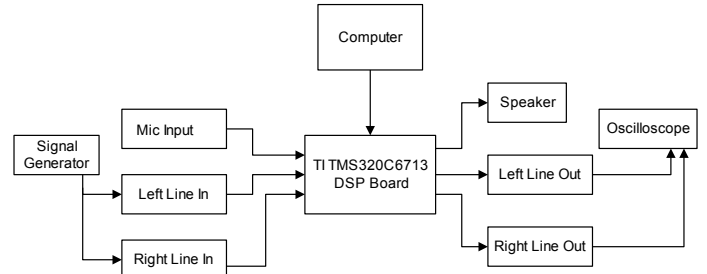


Fig. 5(a) Real-time DSP system setup.

```
float xL[1]={0.0};
float xR[1]={0};
float yL[1]={0.0};
float yR[1]={0.0};
interrupt void c_int11()
{
    float lc; /*left channel input */
    float rc; /*right channel input */
    float lcnew; /*left channel output */
    float rcnew; /*right channel output */
    int i;
    //Left channel and right channel inputs
    AIC23_data.combo=input_sample();
    lc=(float) (AIC23_data.channel[LEFT]);
    rc=(float) (AIC23_data.channel[RIGHT]);
    // Insert DSP algorithm below
    xL[0]=lc; /* Input from the left channel */
    xR[0]=rc; /* Input from the right channel */
    yL[0]=xL[0]; /* simplest DSP equation for the left
channel*/
    yR[0]=xR[0]; /* simplest DSP equation for the
right channel*/
    // End of the DSP algorithm
    lcnew=yL[0];
    rcnew=yR[0];
    AIC23_data.channel[LEFT]=(short) lcnew;
    AIC23_data.channel[RIGHT]=(short) rcnew;
    output_sample(AIC23_data.combo);
}
```

Fig. 5(b) Program segment for verifying input and output.

#### B. Real-time Applications

A configuration for adaptive noise cancellation is shown in Fig. 6, where the primary signal is a generated sinusoid and corrupted internally by the sensed reference noise from the ADC. The reference noise can be set as a tonal noise fed via a function generator or another noise source using the microphone and amplifier. The output signal  $e(n)$  (difference between the corrupted signal and the adaptive filter output) retains a desired frequency as the one from the internal digital oscillator; i.e. the output signal is the recovered sine wave when compared with the original sinusoid and verified from oscilloscope display.

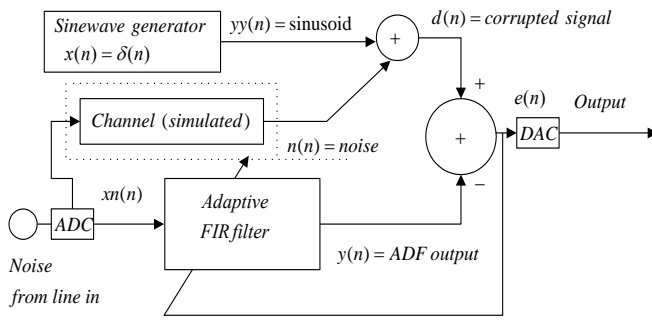


Fig. 6(a) Block diagram of Noise canceller using the adaptive FIR filtering.

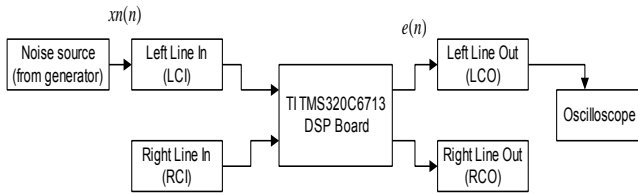


Fig. 6(b) Real-time noise canceller using the adaptive technique and DSP board.

Another interesting and challenging application is the acoustic echo cancellation. The basic telephone echo cancellation is described in Fig. 7. As depicted in Fig. 7, the incoming signal is  $x_B(n)$  from speaker B, which may be leaked to the outgoing signal  $x_A(n)$  from speaker A; that is,  $d_A(n) = x_A(n) + \bar{x}_B(n)$ . If the leakage  $\bar{x}_B(n)$  returns back to the speaker B, it becomes an annoying echo. To prevent the echo, the adaptive filter at speaker A site uses the incoming signal from speaker B as inputs and makes its output approximate to the leaked speaker B signal by adjusting its filter coefficients; the estimated echo  $y_A(n) \approx \bar{x}_B(n)$  is then subtracted from the outgoing signal, thus producing a signal that only represents speech A; that is,  $e_A(n) \approx x_A(n)$ . As a result, the echo from speaker B is removed.

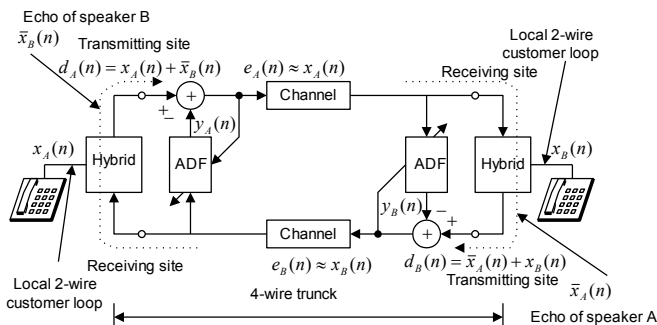


Fig. 7 Adaptive echo cancellers.

The system implementation is shown in Fig. 8, where the objective is to cancel the acoustic feedback instead of a telephone echo. The principle of using adaptive filters remains the same. For comparative purposes, an adaptive filter at the speaker A site (DSP board 1) is used, which cancels the acoustic feedback from speaker B. The second DSP board does not use an adaptive filter in order to show the existence of the acoustic feedback (echo), which can be monitored from Left Line Out (LCO2) via an oscilloscope. The acoustic feedback cancellation effect can be verified at the speaker B site from Right Line Out (RCO1), and the cancellation result is satisfactory. Notice that two additional audio sources (speakers) for microphone A and microphone B are required.

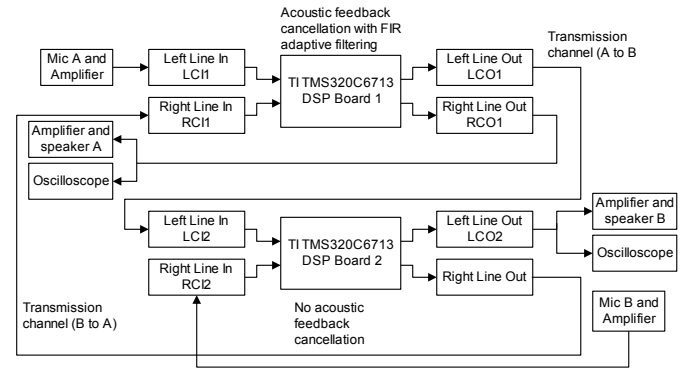


Fig. 8 Acoustic feedback noise cancellation.

## IV. CONCLUSION

In this paper, the novel adaptive real-time noise canceller and echo canceller are presented and implemented, where the proposed adaptive cancellation structure is developed based on the LMS algorithm and FIR adaptive filtering and tested via a real speech waveform. The noise cancellation results are very satisfactory. The challenging adaptive cancellers are also realized in real-time implementations with TI TMS320C6713 DSP boards, in which the floating-point DSP boards demonstrate the accurate iterated and adaptive solutions using C.

## REFERENCES

- [1] L. Tan, J. Jiang. *Digital Signal Processing: Fundamentals and Applications*, 2<sup>nd</sup> edition, Elsevier/Academic Press, New York, 2013.
- [2] E. Ifeachor and B. Jervis. *Digital Signal Processing, A Practical Approach*, Prentice-Hall Publishing, 2002.
- [3] Joyce Van de Vegte. *Fundamentals of Digital Signal Processing*, Prentice-Hall Publishing, 2002.
- [4] L. Tan, J. Jiang, "Teaching Advanced Digital Signal Processing with Multimedia Applications in Engineering Technology Programs," 2011 Proceedings of the American Society for Engineering Education, June 2009.
- [5] H.Wu, S. Kuo, "Teaching Challenge in Hands-on DSP Experiments for Night-School Students," EURASIP Journal on Advances in Signal Processing, Vol. 2008, Article ID 570896, June 2008.

- [6] C. Wicks, "Lessons Learned: Teaching Real-Time Signal Processing," IEEE Signal Processing Magazine, pp. 181-185, November 2009.
- [7] Texas Instruments, *TMS320C6x CPU and Instruction Set Reference Guide*, Literature ID# SPRU 189C, Texas Instruments, Dallas, Texas, 1998.
- [8] Texas Instruments, *Code Composer Studio: Getting Started Guide*, Texas Instruments, Dallas, Texas, 2001.
- [9] L. Tan, J. Jiang, *Analog Signal Processing and Filter Design*, Linus Publications, 2009.



**Jean Jiang** (M'92) received the B.S. and M.S. degrees in electrical engineering from the Southeast University, Nanjing, China, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from the University of New Mexico, Albuquerque, in 1992. She was a Full Professor with DeVry University, Decatur, Georgia. She is currently with the College of Engineering and Technology, Purdue University North Central, Westville,

Indiana. Her research interests include the areas of digital signal processing, active noise control and control systems. She coauthored two textbooks: *Digital Signal Processing: Fundamentals and Applications, Second Edition* (Elsevier, 2013) and *Analog Signal Processing and Filter Design* (Linus Publications, 2009).